

Concise Radiometric Calibration Using The Power of Ranking

Han Gong*

h.gong@uea.ac.uk

Graham D. Finlayson*

g.finlayson@uea.ac.uk

Maryam M. Darrodi

School of Computing Sciences

University of East Anglia

Norwich, UK

* denotes co-first author

Abstract

Compared with raw images, the more common JPEG images are less useful for machine vision algorithms and professional photographers because JPEG-sRGB does not preserve a linear relation between pixel values and the light measured from the scene. A camera is said to be radiometrically calibrated if there is a computational model which can predict how the raw linear sensor image is mapped to the corresponding rendered image (*e.g.* JPEGs) and vice versa. This paper begins with the observation that the rank order of pixel values are mostly preserved post colour correction. We show that this observation is the key to solving for the whole camera pipeline (colour correction, tone and gamut mapping). Our rank-based calibration method is simpler than the prior art and so is parametrised by fewer variables which, concomitantly, can be solved for using less calibration data. Another advantage is that we can derive the camera pipeline from a single pair of raw-JPEG images. Experiments demonstrate that our method delivers state-of-the-art results (especially for the most interesting case of JPEG to raw).

1 Introduction

Many computer vision algorithms (*e.g.* photometric stereo [18], photometric invariants [11], shadow removal [14], and colour constancy [2]) assume that the captured RGBs in images are linearly related to the actual scene radiance. However, the imaging pipeline in a digital camera is necessarily non-linear in order to produce perceptually-pleasing photos rather than its physically-meaningful counterparts. In this paper, we present a new rank-based radiometric calibration method which solves for the bi-directional mappings between the camera's RAW responses and the rendered RGBs produced by digital camera.

There is prior art in this field which models the pipeline with a large number of parameters (up to several thousand [5]) which both means a large corpus of data is required to uncover the pipeline and that there is at least tacitly the premise that the underlying pipeline is quite complex. The key insight in our approach is that post-colour correction (a 3×3 matrix correction) the linear corrected raw RGBs are to the greatest extent in the same rank order as the final rendered RGBs. Building on this insight, we develop a simple rank-based

radiometric calibration model that “solves for” the camera pipeline with many fewer parameters and concomitantly needs much less training data.

In Fig. 1, we illustrate a conventional image reproduction pipeline that holds for many cameras [15]. An exemplar raw image, Fig. 1a, is mapped by a 3×3 colour correction matrix to give the image shown in Fig. 1b. The colour correction matrix implements several processing steps (*e.g.* illumination correction [4, 20], display RGB mapping [1], and colour preference adjustments [20]). It is well-known that a display device cannot display all captured image colours that some RGBs will fall outside the RGB cube after mapping (*e.g.* the pixels marked in purple in Fig. 1b). We therefore need gamut mapping, *e.g.* [5, 12, 15], to bring the colours back inside the cube as shown in Fig. 1c. Finally, the gamut mapped image is tone mapped to arrive at the final rendered output [4, 15, 20] shown in Fig. 1d. Tone mapping accounts for the display non-linearity [1], dynamic range compression and some aspects of preference [21].

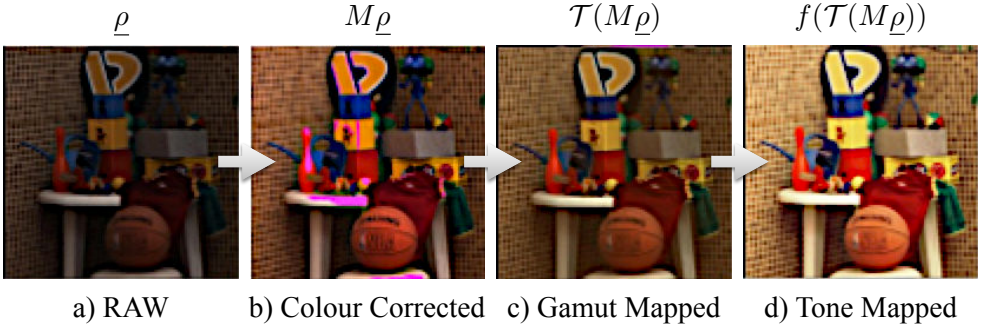


Figure 1: a) a RAW input image is colour corrected to give image b). Non-displayable colours are highlighted in purple pseudo colour. Gamut mapping, in step c), brings colours within gamut. Finally, in d), a tone mapping step results in the final rendered image.

The colour processing pipeline – for cameras, in general, can be written as Eqn. 1.

$$\underline{P} = \underbrace{f(\Gamma(M\underline{\rho}))}_{(1a)} = \underbrace{\Gamma(f(M\underline{\rho}))}_{(1b)} \approx \underbrace{\text{LUT}(\underline{\rho})}_{(1c)} \quad (1)$$

Here and throughout this paper, $\underline{\rho}$ denotes a camera RAW and \underline{P} refers to its rendered RGB counterpart. Respectively, the 3×3 correction matrix, gamut mapping and tone mapping are denoted by the matrix M and the functions $\Gamma()$ and $f()$. The function $f()$ can implement a single or three per-channel tone curves. Because gamut mapping only implements a small change in comparison with colour and tone mapping steps, the order of gamut mapping and tone mapping may be switched (Eqn. 1b & c), a property that we exploit in this paper. Equally, we can also roll all three processing steps into one and directly solve for a 3D LUT (Look-Up-Table) that maps RAW to rendered counterparts. This LUT function is denoted $\text{LUT}()$ [16] in Eqn. 1c. Readers may refer to the top row of Fig. 1 to link each mathematical function to our example processed image.

In radiometric calibration, given a set of $\underline{\rho}$ and \underline{P} , one seeks to solve for the parametrised pipeline parts (*e.g.* M , $\Gamma()$, $f()$ and $\text{LUT}()$). A disadvantage of the current best performing methods is that a great deal of data may be required to fit their assumed models. In Eqns. 1a

and 1b, the gamut mapping step could be modelled by 1000s of Radial Basis functions [5, 15, 16] and in Eqn. 1c, the deployed LUT could also have several thousand control points.

Our proposed method begins with the simple observation [8] that, assuming the gamut mapping step makes small changes to image colours, we expect the rank ordering of the rendered \underline{P} to be the same as \underline{p} multiplied by the correction matrix M (because a tone curves are always monotonically increasing). Suppose that two rendered (JPEG) responses – in the 1st red colour channel – are denoted P_1^a and P_1^b ; and that $P_1^a > P_1^b$. The rank order of two corresponding raw red channel measurements post colour correction is written as $M_1 \underline{p}^a > M_1 \underline{p}^b$ (where M_1 denotes the first row of M and \underline{p}^a and \underline{p}^b are a pair of raw RGBs). Rewriting: this implies that $M_1(\underline{p}^a - \underline{p}^b) > 0$ which mathematically defines a half-space constraint. If we visualise the row vector M_1 as a point in 3-space then this inequality – which we call a ranking constraint – forces the point to be located in one half of 3-space but not the other. Because we have multiple pixels, each pair of pixels (2 raw and 2 JPEG RGBs) generates a half space constraint and intersecting all these constraints delimits the region in which M_1 must lie. Our experiments demonstrates that a small numbers of patches suffices to estimate M accurately. Once we have M we then find the best rank preserving tone curves $f()$. At this stage, only using M and $f()$ we have a tolerable approximation of the pipeline. Indeed, we argue that our construction of M and $f()$ also incorporates, to a first order, gamut mapping. Now we adopt (Eqn 1c) and find a 125-parameter LUT to “mop up” any remaining errors due to gamut mapping (higher order terms).

2 Related work

Using the pipeline form Eqn 1b, Chakrabarti *et al.* [5] first solve for M and $f()$ (in a least-squares sense) in iteration and then solve directly for $\Gamma()$. In their approach, $f()$ is constrained to be a 7th order monotonic polynomial. They model $\Gamma()$ with the radial basis function (RBF) method of [15] where several thousands of RBFs are potentially used. A restriction of the above calibration is presented in [4] where the gamut mapping $\Gamma()$ is ignored. This less general model works tolerably well on many real pairs of raw and rendered images and this is a point we will return to later in this paper. In either version ([5] or [4]), the coupled nature of the minimization means that a global minimum is not guaranteed to be found. Thus, a random start search is incorporated – multiple minimisations are carried out – in order to find their best parameters. Kim *et al.* [15] solve for the pipeline in the form of Eqn. 1a and makes additional assumptions to decouple the optimization. They assume that images of the same scene are captured with respect to two or more exposures and their $\Gamma()$ is a multi-thousand set of RBFs. Regarding solving for $f()$, Debevec *et al.* [6] showed how relating corresponding pixels under known exposure differences suffices to solve for $f()$ (assuming there is no gamut mapping step). Importantly, in [15], it was argued that for the set of desaturated pixels (*i.e.* RAWs far from the RGB cube boundary) the gamut mapping step has little or no effect and can be ignored. Relative to this assumption, $f()$ can be solved using the Debevec method. Given $f()$ then the colour correction matrix M can be found (again using desaturated pixels). Though, for typical capture conditions, *e.g.* for most mobile phones, multiple exposures are not available and so the requirement that multiple exposures are needed is a weakness in this method. Finally, in [15] a gamut mapping RBF network is “trained”. Of course, if a large number of radial basis functions are used to model gamut mapping (as proposed in [15] or [5]) then solving for $\Gamma()$ requires a large corpus of data. Further the application of gamut mapping is expensive and its parametrisation is large.

In [16] it was shown that is possible to ignore the underlying structure of the colour processing pipeline and directly solve for the best 3D surjective function – implemented as a LUT that maps the RAWs to rendered RGBs (Eqn. 1c). Finally, in [17], a method is presented for solving for $f()$ by examining the edge distribution in an image. This method has the advantage that the method works for a single image (no need for multiple exposures) but the disadvantage that the method is sensitive to processing steps such as image sharpening which is used extensively in mobile phone image processing.

3 The rank-based method

As the reader shall see, to make the rank-based method work we need to assume that the gamut mapping step $\Gamma()$ only makes small adjustments to colour. In fact our assumption is more nuanced. We assume that – to a first order – gamut mapping can mostly be implemented as an affine transform and that this affine transform can be folded into the colour correction matrix M and the monotonically increasing tone mapping functions $f()$.

3.1 Gamut Mapping as an Affine Transform

In Eqn. 1b, gamut mapping is applied when, after colour correction, colours are mapped outside the colour cube and become non-displayable. Let us use a Taylor expansion to model $\Gamma()$ around a point \underline{a} inside the gamut:

$$\Gamma(M\underline{p}) \approx \Gamma(\underline{a}) + J(\underline{a})(\underline{p} - \underline{a}) \quad (2)$$

where J is the 3×3 Jacobian (matrix of derivatives of Γ). Not only does Eqn. 2 show that, to a first approximation, gamut mapping is an affine transform it is also one of the gamut mapping algorithms proposed in [12]. In particular, [12] solves, with good results, for the best affine transform that maps image colours inside the gamut and which are, simultaneously, close to the non-gamut mapped colours:

$$\min_{\underline{T}, \underline{o}} \sum_i ||TM\underline{p}_i + \underline{o} - M\underline{p}_i||^2 \text{ s.t. } \underline{0} \leq TM\underline{p}_i + \underline{o} \leq \underline{1} \quad (3)$$

In Eqn. 3, T and \underline{o} are respectively a 3×3 matrix and 3×1 offset vector defining the affine gamut mapping algorithm. The 3-vectors of 0s and 1s are denoted $\underline{0}$ and $\underline{1}$. Eqn. 3 is solved directly by Quadratic Programming [13]. The gamut mapping shown in Fig. 1c is the result of solving Eqn. 3.

Here, we make two important remarks about affine gamut mapping: 1) Gamut mapping and colour correction combined can be represented by the single affine transform: 3×3 matrix TM and offset \underline{o} ; 2) It follows that the rank-based method presented in the next section will actually solve for TM . The offset term can be incorporated directly in $f()$ (since an offset does not change ranking).

3.2 Rank-based estimation for colour correction

Let us denote the k^{th} row of M as M_k , let us assume that given two colour corrected RAWs, $M_k \underline{p}^a$ and $M_k \underline{p}^b$ that the rank order is the same as for the corresponding rendered RGBs:

$$P_k^a > P_k^b \Rightarrow M_k \underline{p}^a > M_k \underline{p}^b \Rightarrow M_k(\underline{p}^a - \underline{p}^b) > 0 \quad (4)$$

Defining the difference vector $\underline{d}^j = \underline{\rho}^a - \underline{\rho}^b$:

$$M_k \underline{d}^j > 0 \quad (5)$$

where it is understood the superscript j denotes the difference vector from the j^{th} of $\binom{n}{2}$ pairs of n image pixel values. Suppose that we have a vector M_k where Eqn. 5 holds, then the inequality cannot be true for $-M_k$. That is Eqn. 5 defines a half plane constraint [3, 8]. The vector \underline{d}^j is perpendicular to the half-plane: any M_k less than 90 degrees to \underline{d}^j is a possible solution. Given multiple difference vectors then we have multiple half-plane constraints which taken together delimit a region in 3-space where M_k must lie. Denoting the half-plane as $\mathcal{H}(\underline{d}^j)$, M_k must satisfy:

$$M_k \in \bigcap_j \mathcal{H}(\underline{d}^j) \quad (6)$$

Let us visualise the computation of M_k using ranking. Without loss of generality let us assume that $M_{k,3} = 1$. We rewrite Eqn. 5 as

$$M_{k,1}d_1^j + M_{k,2}d_2^j + d_3^j > 0 \quad (7)$$

If $[a \ b \ c]$ is a solution to Eqn. 6, then $[a/c \ b/c \ c/c]$ for Eqn. 7 is also true since $M_{k,1} = a/c$ and $M_{k,2} = b/c$. Solutions for $[M_{k,1}, M_{k,2}]$ lie on one side of the line, *i.e.* the 3D half-space constraints maps directly to a 2D half-plane constraint. Or, if we consider the whole set of collations, the cone in 3D, defined by Eqn. 6, maps to a 2D convex region [7]. Denoting half-planes as $\mathcal{P}(\underline{d}^j)$ we, equivalently, solve for

$$[M_{k,1}, M_{k,2}] \in \bigcap_j \mathcal{P}(\underline{d}^j) \quad (8)$$

The intersection problem of Eqn. 8 is easily visualised. In Fig. 2a we show the intersection of 4 half plane constraints and indicate the solution set where M_k must lie.

We solve for M one sensor channel at a time. Empirically, we have to be careful not to generate too many half planes. In our experiment, we generate half-planes from all pairs of up to 50 randomly selected unique RAWs, generating 2450 half-planes. Due to noise or small deviations in real camera data, it is likely that no common intersection can be found that satisfy every half-planes constraint. To solve this problem, we generate 100,000 unit length vectors that are uniformly distributed on the surface of the unit sphere [19], which is visualised in Fig. 2b. With respect to this sampling, the furthest distance between any point and its nearest neighbour is less than 1.15 degrees. So, the orientation of the rows of M are found to this accuracy. For each point on the sphere (*i.e.* a possible row of M_k), we count how many half-space constraints are satisfied. The point on the unit sphere that satisfies most half-plane constraints – or the median of multiple points if there is a tie – defines M_k . To make our approach robust, we find randomly select 50 colours 25 times and for each trial find the best M_k . Overall, we find the M that places all the corresponding raw and rendered image RGBs in the most similar rank order. That is, if we plot the mapped raw red responses, for example, against the rendered red JPEG corresponding values then the graph should be a monotonically increasing function. How well a monotonically increasing function fits our data can be used to judge the efficacy of each M .

Ranking can only estimate M up to an unknown scaling of its rows. Suppose for a rendered achromatic RGB $\underline{\rho}^A = [0.5 \ 0.5 \ 0.5]^\top$ and the corresponding raw $\underline{\rho}^A = [a \ b \ c]^\top$,

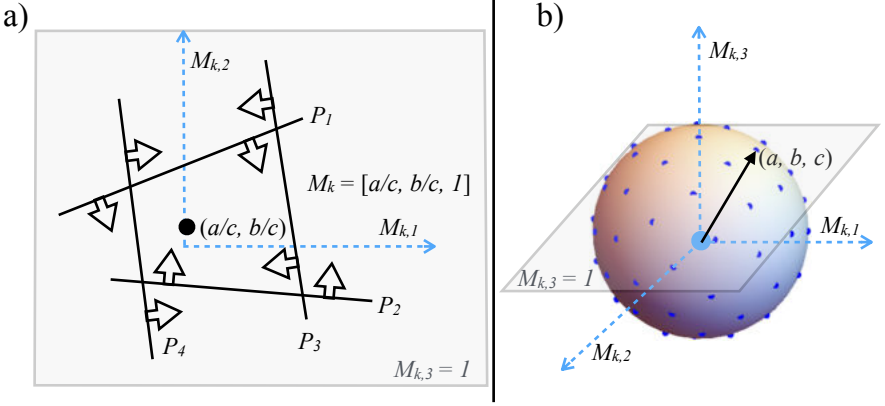


Figure 2: a) The region where 4 half-plane constraints intersect delimit the region where $[M_{k,1}, M_{k,2}]$ must lie where the black point is a feasible solution. b) On an unit sphere, each vector represented by the origin and a blue surface point is a probe for a possible solution (e.g. the black arrow). All 3D points and constraints are projected to a 2D plane $M_{k,3} = 1$.

we apply: $0.5\mathbf{diag}(M\rho^A)^{-1}M\rho^A = [0.5 \ 0.5 \ 0.5]^\top$ where $\mathbf{diag}()$ places a vector along the diagonal of a diagonal matrix. After this step, $M \leftarrow 0.5\mathbf{diag}(M\rho^A)^{-1}M$ maps achromatic colours correctly. Because $M \leftarrow DM$ (where in the example, $D = 0.5\mathbf{diag}(M\rho^A)^{-1}$) we might also solve for D in a least-squares sense by including all colours indexed i close to the achromatic axis: $\min_D \sum_i \|DM\rho^i - \underline{P}^i\|$ (our experiment does not include this additional step).

3.3 Rank-preserving optimization of tone curves

We now solve for the optimal per-channel tone curves which map colour corrected RAWs to corresponding rendered RGBs. Let us denote the i^{th} colour corrected RAW and rendered RGB pixel pairs for the k^{th} channel as $(\rho_{k,i}, P_{k,i})$. Then, the k^{th} -channel rank-preserving tone curve $f_k()$ is optimised as a 7th order monotonic and smooth polynomial function as follows:

$$\min_{f_k()} \sum_i \|f_k(\rho_{k,i}) - P_{k,i}\|^2 + \lambda \int_t \|f_k''(t)\|^2 dt \quad \text{s.t. } f_k'(t) \geq 0. \quad (9)$$

where the first term is for data fitness, the second term is for curve smoothness and λ is a small weight (e.g. 10^{-5}). This polynomial fitting is solved by Quadratic Programming [13]. In this paper, we further denote the combination of all 3-channel mappings $f_{1-3}()$ as $f()$.

3.4 Gamut correction step

As argued previously, we propose that $f(M\rho)$ has the expressive power to implement colour correction, tone correction and gamut mapping (to the first order in a Taylor expansion). However, we wish to add a further gamut mapping step for the higher order terms. But, since our hypothesis is that much of the gamut mapping will have been accounted for we are going to adopt a simple small parameter solution. Further, this additional correction is going

to be carried out at the end of the process, we adopt Eqn. 1b. Specifically, we find a $5 \times 5 \times 5$ LUT by using lattice regression [10] that minimises $\min_{LUT()} \sum_i ||LUT(f(M\underline{\rho}_i)) - \underline{P}_i||^2$.

3.5 Rank-based recovery of raw

Suppose we wish to map rendered RGBs to RAWs. Using the method presented in Section 3.2, M has already been solved in the RAW-to-JPEG forward estimation phrase. Now, in a least-squares optimal way, we use the same polynomial fitting method (Eqn. 9) to find f^{-1} by optimising $\min_{f^{-1}()} \sum_i ||f^{-1}(\underline{P}_i) - M\underline{\rho}_i||$. Finally, we solve for the backward LUT() by optimising $\min_{LUT()} \sum_i ||LUT(M^{-1}f^{-1}(\underline{P}_i)) - \underline{\rho}_i||$ where the LUT is fitted by a $5 \times 5 \times 5$ lattice regression [10].

3.6 Parameter counting

Assuming we solve for 3 independent tone curves then our method requires 9 (for M) + 24 (for $f()$) + 125 (for $\Gamma()$) = 158 parameters which is significantly less (even an order of magnitude less) than [5, 15, 16].

4 Evaluation

Our evaluation is based on the most challenging dataset that we have encountered: [5] which contains the RAW/JPEG intensity pairs of 140 colour checker patches viewed under multiple viewing conditions. Specifically, the colour chart is captured by 8 cameras (3 of which are JPEG-only) and under 16 illuminants across many different exposures.

Below, we carried out the same experiment described in [5]. We are interested in validating whether our method, with much reduced number of parameters can produce, similar or even better results compared with [5]. We evaluate both RAW-to-JPEG and JPEG-to-RAW. The dataset [5] captures a sort of “worst-case” viewing conditions. Normally, when we capture a picture there is a single prevailing illuminant colour. In the dataset of Chakrabarti *et al.*, all camera processing parameters are turned off and then the same reflectances are viewed under multiple coloured lights. As Forsyth observed [9], the reddest red camera response cannot be observed under a blue light. And, then he exploited this observation to solve for the colour of the light. Yet, in this dataset the reddest red, the greenest green and the bluest blue can all appear simultaneously in the same image. Practically, we believe the need to divine a pipeline for the all lights all surfaces case means the prior art pipelines are probably more complex than they need to be.

As described in [5], for each camera, we estimate the parameters of a calibration model using different subsets of available RAW-JPEG pairs. For each subset and a selected camera, the root mean-squared error (RMSE) between the prediction and ground truth is validated by using all available RAW-JPEG pairs. Table 1a and Table 1c shows the RAW-to-JPEG mapping error (where pixel intensities are coded as integers in the interval [0, 255]). In the table, *Prob* denotes the Chakrabarti method (with several thousands parameters) and *RB* the rank-based method with 158 parameters. We found that our forward errors are close to the results of [5], especially for the condition of less than 3 illuminants which are more likely to occur in the real world. Evidently, for the many illuminant case the prior art has a small advantage. Remembering that JPEGs are coded as integers in [0, 255] the RMSE is typically

1 or less (for *RB* compared to *Prob*). Practically, when the “fits” are viewed visually (by looking at images) it has hard to see the difference between the two methods.

For computer vision, we are more interested in the performance of JPEG-to-RAW mapping which is shown in Table 1b and Table 1c. In [5], a probabilistic framework for mapping rendered RGB to raw was presented. Here we take their mean estimates as the most likely RAW predictions. We found that our method generally reduces the errors of [5] by $\sim 34\%$. Our supplementary material also includes the additional experiment results compared with “[4] + our LUT” for interested readers.

The reader might be interested why our simple method seems to work so well going from rendered to raw (better than [5]) but not quite as well as the prior art in the forward direction (albeit visually almost indistinguishable). Our hypothesis here is that the LUT in the forward direction is applied post the tone curve. This curve (at least for dark values) has a very high slope and, consequently, the coarsely quantised $5 \times 5 \times 5$ LUT cannot capture gamut mapping well. Yet, in the reverse direction (JPEG to RAW) the LUT is applied in linear raw where a coarse uniform quantisation is more justified.

5 Calibration with small numbers of parameters

We wished to visually validate our claim that we can calibrate with few parameters. We took 4 RAW+JPEG pairs (for different cameras) from [4]. We then uniformly selected 140 corresponding pixels from the RAW and JPEG. We solved for all the 158 parameters in our rank-based method. We then applied our model to the rest of the image. The result of this experiment for 4 images (JPEG-to-RAW) is shown in Fig. 3.

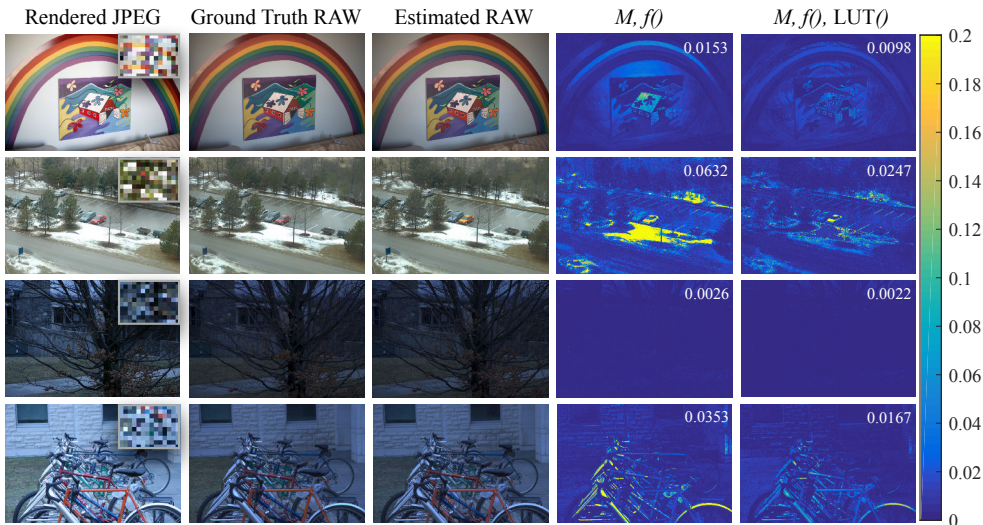


Figure 3: Visualisation of one-shot radiometric calibration through a simulated 140-patch colour checker, shown at the top-right corner of each Rendered JPEG image. The error maps in the 4th and 5th columns respectively visualise the per pixel RMSE for our rank-based method with & without the gamut mapping LUT. The RMSE of each whole image is shown at the top-right corner of each error map. All raw images are shown with a 0.5 gamma.

a) RAW-to-JPEG		Uniform 8K		10 Exp. 1 Illu.		10 Exp. 2 Illu.		4 Exp. 4 Illu.	
Camera		Prob	RB	Prob	RB	Prob	RB	Prob	RB
Canon_EOS_40D		1.84	2.56	9.79	10.10	7.53	4.13	4.06	5.87
Canon_G9		2.17	3.70	6.51	6.20	3.41	5.48	3.09	4.79
Canon_PowerShot_S90		2.44	3.24	4.88	4.52	3.58	4.34	3.40	4.04
Nikon_D7000		1.72	4.03	8.05	10.03	3.32	5.39	26.06	6.48
Panasonic_DMC-LX3		1.65	3.65	7.33	8.70	5.25	4.56	3.05	7.98
				8 Exp. 4 Illu.		4 Exp. 6 Illu.		8 Exp. 6 Illu.	
Camera				Prob	RB	Prob	RB	Prob	RB
Canon_EOS_40D				2.91	4.13	3.60	4.11	2.25	3.61
Canon_G9				2.79	5.48	3.12	4.74	2.77	4.67
Canon_PowerShot_S90				2.95	4.34	3.27	3.70	2.75	3.93
Nikon_D7000				2.41	5.39	2.77	5.04	1.92	4.95
Panasonic_DMC-LX3				2.77	4.56	2.94	4.26	2.33	4.14
b) JPEG-to-RAW		Uniform 8K		10 Exp. 1 Illu.		10 Exp. 2 Illu.		4 Exp. 4 Illu.	
Camera		Prob	RB	Prob	RB	Prob	RB	Prob	RB
Canon_EOS_40D		0.079	0.060	0.085	0.072	0.080	0.064	0.075	0.072
Canon_PowerShot_G9		0.126	0.075	0.143	0.104	0.120	0.079	0.120	0.082
Canon_PowerShot_S90		0.065	0.052	0.073	0.058	0.069	0.074	0.066	0.057
Nikon_D7000		0.143	0.090	0.543	0.123	0.140	0.098	0.229	0.108
Panasonic_DMC-LX3		0.082	0.058	0.090	0.072	0.082	0.063	0.073	0.071
				8 Exp. 4 Illu.		4 Exp. 6 Illu.		8 Exp. 6 Illu.	
Camera				Prob	RB	Prob	RB	Prob	RB
Canon_EOS_40D				0.071	0.064	0.077	0.065	0.069	0.063
Canon_PowerShot_G9				0.121	0.079	0.126	0.076	0.126	0.080
Canon_PowerShot_S90				0.069	0.074	0.063	0.059	0.066	0.058
Nikon_D7000				0.144	0.098	0.147	0.094	0.143	0.101
Panasonic_DMC-LX3				0.077	0.063	0.074	0.060	0.077	0.064
c) Uniform 8K (JPEG-Only Camera Test)						RAW-to-JPEG		JPEG-to-RAW	
Camera	Raw Proxy					Prob	RB	Prob	RB
FUJIFILM_J10	Panasonic_DMC-LX3					10.43	11.51	0.279	0.077
Galaxy_S_III	Nikon_D7000					11.34	13.13	0.114	0.074
Panasonic_DMC_LZ8	Canon_PowerShot_G9					8.85	12.23	0.146	0.085

Table 1: RMSE between ground truth and prediction for bidirectional RAW and JPEG conversions: Prob denotes [5] and RB is our rank-based method. “Exp.” and “Illu.” are respectively short for “Exposure” and “Illuminant”. “Raw Proxy” is the camera used to capture raw for the camera which does not support raw image capturing.

6 Conclusion

In this paper we have shown how the rank order of image responses is a powerful tool for solving for the individual steps in a camera processing pipeline (colour correction, gamut and tone mapping). A simple ranking argument, relating colour corrected RAWs to corresponding rendered RGBs suffices to solve for the colour correction matrix. Then, the rank-preserving tone map is found and, finally, a simple gamut correction step is derived. Compared with the prior art, our rank-based method requires the fewest assumptions and delivers state-of-the-art radiometric calibration results.

References

- [1] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet:srgb. In *IS&T and SID's 4th Color And Imaging Conference*, pages 238–245. 1995.
- [2] Kobus Barnard, Lindsay Martin, Adam Coath, and Brian V. Funt. A comparison of computational color constancy algorithms. ii. experiments with image data. *IEEE Transactions on Image Processing*, 11(9):985–996, 2002.
- [3] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, 3rd ed. edition, 2008.
- [4] Ayan Chakrabarti, Daniel Scharstein, and Todd Zickler. An empirical camera model for internet color vision. In *British machine Vision Conference*, volume 1, page 4, 2009.
- [5] Ayan Chakrabarti, Ying Xiong, Baochen Sun, Trevor Darrell, Daniel Scharstein, Todd Zickler, and Kate Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *Transactions on pattern analysis and machine intelligence*, 36(11):2185–2198, 2014.
- [6] Paul Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, pages 369–378, 1997.
- [7] G.D. Finlayson. Color in perspective. *Transactions on Pattern analysis and Machine Intelligence*, pages 1034–1038, October 1996.
- [8] Graham Finlayson, Maryam Mohammadzadeh Darrodi, and Michal Mackiewicz. Rank-based camera spectral sensitivity estimation. *Journal of the Optical Society of America A*, 33(4):589–599, 2016.
- [9] David A Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–35, 1990.
- [10] Eric K. Garcia, Raman Arora, and Maya R. Gupta. Optimized regression for efficient function evaluation. *IEEE Transactions on Image Processing*, 21(9):4128–4140, 2012.
- [11] Theo Gevers, Graham D. Finlayson, and Raimondo Schettini. Color for image indexing and retrieval. volume 94, pages 1–3, 2004.

- [12] J. Giesen, E. Schuberth, K. Simon, P. Zolliker, and O. Zweifel. Image-dependent gamut mapping as optimization problem. *Image Processing, IEEE Transactions on*, 16(10): 2401–2410, Oct 2007.
- [13] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], 1981.
- [14] Han Gong and Darren Cosker. Interactive removal and ground truth for difficult shadow scenes. *Journal of the Optical Society of America A*, 33(9):1798–1811, Sep 2016.
- [15] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Susstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012. ISSN 0162-8828.
- [16] Hai Ting Lin, Zheng Lu, Seon Joo Kim, and Michael S. Brown. Nonuniform lattice regression for modeling the camera imaging pipeline. In *European Conference on Computer Vision*, volume 7572 of *Lecture Notes in Computer Science*, pages 556–568. Springer, 2012.
- [17] S. Lin, Jinwei Gu, S. Yamazaki, and Heung-Yeung Shum. Radiometric calibration from a single image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–938–II–945 Vol.2, June 2004.
- [18] A.P. Petrov. On obtaining shape from color shading. *Color research and application*, 18(4):236–240, 1993.
- [19] E.A. Rakhmanov, E.B. Saff, and Y.M. Zhou. Minimal discrete energy on a sphere. *Mathematical Research letters*, pages 647–662, 1994.
- [20] R. Ramanath, W.E. Snyder, Y. Yoo, and M.S. Drew. Color image processing pipeline. *Signal Processing Magazine, IEEE*, 22(1):34–43, 2005.
- [21] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Transaction on Graphics*, 21(3):267–276, July 2002.